ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Υπηρεσία παρακολούθησης θερμοκρασίας σε Server Rooms

Υπηρεσίες Υπολογιστικού Νέφους

Αναστασιάδης Παναγιώτης 22101
Καραβιβέρη Αλεξάνδρα 22103

20 Σεπτεμβρίου 2023

Πίνακας Περιεχομένων

Εισαγωγή

Τα data centers αποτελούν, σήμερα, κρίσιμες υποδομές για τους οργανισμούς στους οποίους ανήκουν και εσωτερικά φιλοξενούν έναν αριθμό από server rooms, τα οποία είναι απαραίτητα για την λειτουργία των υπηρεσιών τους. Συνεπώς, η διατήρηση της φυσιολογικής θερμοκρασίας στους συγκεκριμένους χώρους, καθώς και στο hardware που αυτά περιέχουν, είναι μια ανάγκη ζωτικής σημασίας για την διασφάλιση της ομαλής λειτουργίας και αδιάκοπης διαθεσιμότητας των υπηρεσιών τους.

Η παρούσα εργασία στηρίζεται στη συγκεκριμένη ανάγκη και μοντελοποιεί μια σύγχρονη λύση που ως ένα σύνολο από διάφορες υπηρεσίες είναι ικανή να εγκατασταθεί σε υποδομές νέφους και να αντιμετωπίσει αποτελεσματικά θέματα θερμοκρασίας.

Με την χρησιμοποίηση αισθητήρα ή αισθητήρων θερμοκρασίας που είναι ικανοί να στέλνουν τις μετρήσεις τους μέσω του πρωτοκόλλου MQTT, η υλοποίηση κάνει τα παρακάτω:

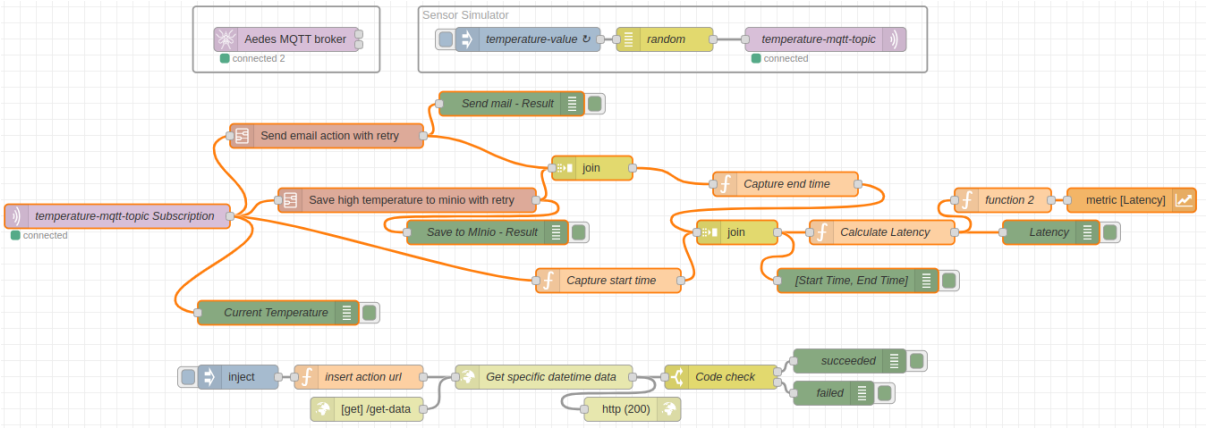
- Λαμβάνει τις μετρήσεις των αισθητήρων μέσω του πρωτοκόλλου MQTT και του σχεδιασμού Publisher/Subscriber (λειτουργεί ως subscriber).
- Επεξεργάζεται την κάθε μέτρηση εντοπίζοντας αυτές που ξεπερνούν ένα ορισμένο κατώφλι.
- Για τις συγκεκριμένες επικίνδυνες θερμοκρασίες ενεργοποιεί, σε πλαίσιο λειτουργίας Function as a Service, δύο δράσεις:
    - Αποστολή σχετικού email ειδοποίησης
    - Αποθήκευση της μέτρησης και της χρονικής σφραγίδας για μελλοντική επεξεργασία.
- Υποστηρίζει μια συνεχή λειτουργία παρακολούθησης της ποιότητας του service που πραγματοποιεί, μετρώντας τη καθυστέρηση από τη στιγμή της λήψης της μέτρησης μέχρι και την ολοκλήρωση των δύο δράσεων.
- Σε περίπτωση μη εκτέλεσης των σχεδιαζόμενων δράσεων, υποστηρίζει λειτουργία επαναλήψεων της προσπάθειας εκτέλεσης τους (Retry Pattern).
- Έχει λειτουργίες REST API, καθώς επιτρέπει την ανάκτηση των αποθηκευμένων δεδομένων-μετρήσεων, υποστηρίζοντας λειτουργίες φιλτραρίσματος (χρονικό διάστημα σε ημερομηνίες).
    - Η ανάκτηση των δεδομένων γίνεται μέσω μιας τρίτης δράσης (και πάλι στο πλαίσιο της FaaS), η οποία ενεργοποιείται για να αναζητήσει τα επιθυμητά δεδομένα στο χώρο αποθήκευσης.

Σε ξεχωριστό αρχείο δίνεται ένα κείμενο που περιέχει τη πιθανή δομή που θα είχε ένα Σύμφωνο Επιπέδου Υπηρεσίας με κάποιον cloud πάροχο.

Στη συνέχεια, αφού σχολιαστεί ο τρόπος με τον οποίον η υλοποίηση ενσωματώνει μοτίβα σχεδιασμού και μηχανισμού, αναφέρονται μερικά πλεονεκτήματα και μειονεκτήματα της εφαρμογής, καθως και η σύγκριση της με σύγχρονες αρχιτεκτονικές.

Αρχιτεκτονική

Στο παρακάτω σχήμα φαίνεται το σχετικό flow της εφαρμογής, έτσι όπως αυτό απεικονίζεται στη πλατφόρμα του Node Red.

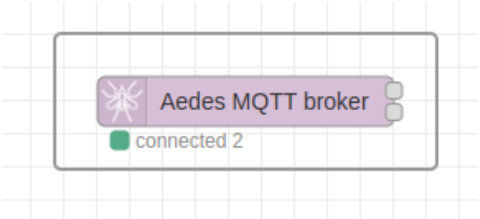


Αναλυτική περιγραφή των components της εφαρμογής:

Node Red

Ο πυρήνας της εφαρμογής είναι το Node Red, καθώς αυτό διαχειρίζεται και ενώνει μεταξύ τους τα διαφορετικά services που απαρτίζουν την εφαρμογή.

Aedes MQTT Broker Node (Pub/Sub Pattern)



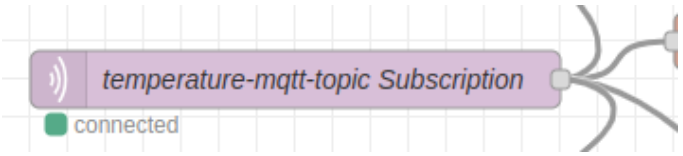
Στο MQTT πρωτόκολλο επικοινωνίας χρειάζεται ένας broker, ο οποίος διαχειρίζεται τα διάφορα κανάλια ή topics και διευθύνει την συνολική επικοινωνία. Για τις ανάγκες της εφαρμογής, χρησιμοποιήθηκε ο Aedes MQTT Broker, ο οποίος στήνεται εξ ολοκλήρου πάνω στον node red server και ακούει στη πύλη **1883**.

Temperature sensor simulator (Pub/Sub Pattern)

Το συγκεκριμένο σύνολο nodes εξομοιώνει τη λειτουργία ενός hardware component (π.χ. ενός arduino microcontroller με συνδεδεμένο πάνω του έναν αισθητήρα θερμοκρασίας), το

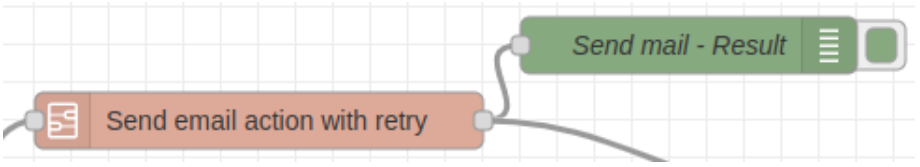
οποίο είναι σχεδιασμένο να τροφοδοτεί μια τιμή θερμοκρασίας κάθε **20 δευτερόλεπτα** στο MQTT topic **"/temperature"**. Για τις ανάγκες της προσομοίωσης, έχει τοποθετηθεί ένας κόμβος που παράγει τυχαίες τιμές θερμοκρασίας, ώστε να προσεγγίζει τη φυσική κατάσταση κατά την οποία η θερμοκρασία θα μεταβάλλεται σε ένα πραγματικό server room.

Subscriber Node (Pub/Sub Pattern)



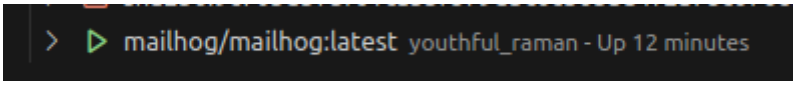
Για τη λήψη και την προώθηση των τιμών θερμοκρασίας, χρησιμοποιείται ένας node που λειτουργεί ως subscriber και ακούει διαρκώς για νέες τιμές στο topic "**/temperature"**.

Sending Mail Alert | Openwhisk Action



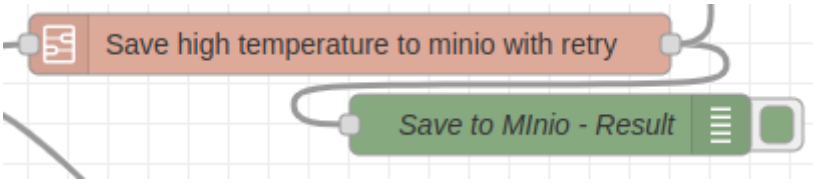
Η βασική λειτουργικότητα της εφαρμογής είναι να εντοπίζει τις θερμοκρασίες εκείνες που έχουν οριστεί ως επικίνδυνες (πάνω από κάποιο ορισμένο κατώφλι) και να ενημερώνει τον χρήστη εγκαίρως. Για το λόγο αυτό, στήθηκε το service του openwhisk, και συγκεκριμένα υλοποιήθηκε ένα action, το οποίο λαμβάνει ως παράμετρο **την τιμή και τη χρονική σφραγίδα της θερμοκρασίας** και στέλνει ένα **σχετικό email συναγερμού** σε μια προκαθορισμένη ηλεκτρονική διεύθυνση ταχυδρομείου και παράλληλα επιστρέφει μήνυμα επιτυχίας ή σφάλματος αναλόγως το αποτέλεσμα.

Mailhog Service



Για τις ανάγκες της εφαρμογής και του σωστού ελέγχου της αποστολής και λήψης των emails, χρησιμοποιήθηκε το service του Mailhog που αποτελεί ένα εργαλείο testing για λειτουργίες email. Συγκεκριμένα, χρησιμοποιήθηκε το σχετικό Docker image του Mailhog, το οποίο προσφέρει ένα γραφικό περιβάλλον που επιτρέπει την άμεση διερεύνηση της ομαλής λειτουργίας της αποστολής των emails κατά το στάδιο της ενεργοποίησης του σχετικού Openwhisk action.

Saving Measurements to Minio | Openwhisk Action



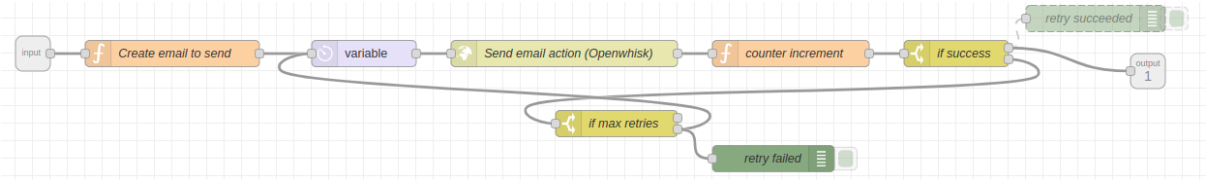
Βασική ανάγκη της εφαρμογής είναι η αποθήκευση των μετρήσεων των επικίνδυνων θερμοκρασιών, ώστε να ωφελήσει μελλοντικά τη διερεύνηση των λόγων που τις

προκάλεσαν. Το συγκεκριμένο action ενεργοποιείται, όπως και το προηγούμενο service (της αποστολής email), κατά την μέτρηση επικίνδυνης θερμοκρασίας. Η τιμή και η χρονική σφραγίδα της θερμοκρασίας αποθηκεύονται στο object storage service της εφαρμογής, το MinIO.

MinIO | Object Storage

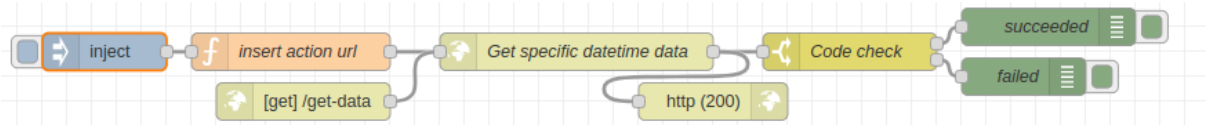
Η εφαρμογή υποστηρίζει την αποθήκευση των δεδομένων των μετρήσεων στο service του MinIO. Αν και τα δεδομένα που αποθηκεύονται από την εφαρμογή δεν δικαιολογούν από μόνα τους την προτίμηση ενός Object Storage Service, η χρησιμοποίηση του διευκολύνει την επέκταση των λειτουργιών και των δυνατοτήτων της εφαρμογής. Αυτό συμβαίνει, καθώς μελλοντικά, θα μπορούσε να προστεθεί η αποθήκευση όγκου εικόνων heatmaps ή video που θα διευκολύνει τον καλύτερο εντοπισμό των υψηλών θερμοκρασιών, καθώς και τις πηγές πρόκλησης αυτών.

Retrying unsuccessful Openwhisk actions (Retry Pattern)



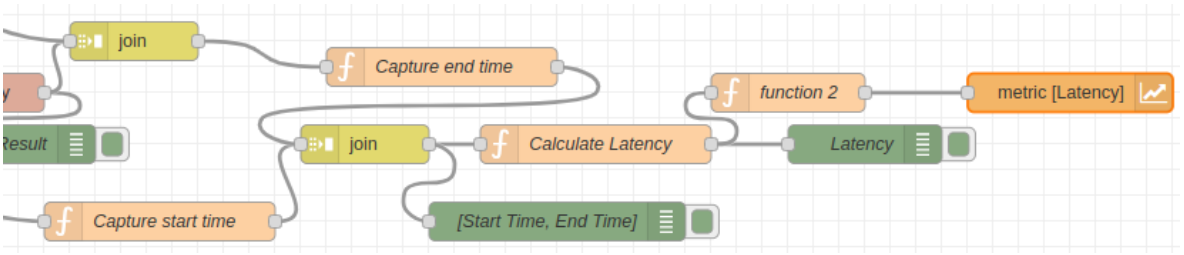
Η εφαρμογή υποστηρίζει διαδικασία επανάληψης ανά ορισμένα από μας χρονικά διαστήματα των παραπάνω Openwhisk actions σε περίπτωση που αυτές ήταν αποτυχημένες. Επιπλέον, είναι δυνατόν μέσα από την εφαρμογή να ορίσουμε και τον αριθμό των συνολικών επαναληπτικών προσπαθειών.

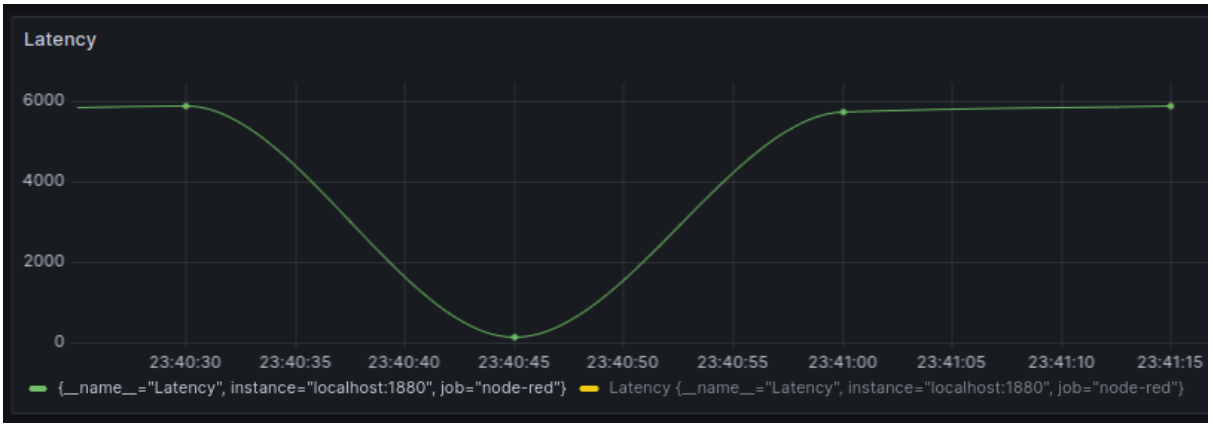
REST API | Fetching Data | Openwhisk Action



Η εφαρμογή πρέπει να υποστηρίζει την εύκολη ανάκτηση των δεδομένων που αποθηκεύονται στο MinIO. Συνεπώς, δημιουργήθηκε στο Node Red ένα σχετικό endpoint “/get-data”, το οποίο επιτρέπει την ανάκτηση των δεδομένων, μέσω σχετικού φιλτραρίσματος (url parameters) που ζητά την αρχική και τελική ημερομηνία από τον χρήστη. Οι ημερομηνίες αυτές αποτελούν το χρονικό διάστημα για το οποίο το API θα φέρει τα σχετικά δεδομένα. Όταν το API λαμβάνει ένα τέτοιο request, ενεργοποιείται το Openwhisk action που θα ανακτήσει τα συγκεκριμένα δεδομένα.

Monitoring Latency and MQTT topics | Prometheus and Grafana





Η εφαρμογή υποστηρίζει τη μέτρηση της καθυστέρησης της βασικής της λειτουργίας που δεν είναι άλλη από την αποστολή email alerts και την αποθήκευση των δεδομένων, όταν προκύπτουν υψηλές θερμοκρασίες. Για το λόγο αυτό, αφού μετρηθεί η καθυστέρηση σε milliseconds (από τη μέτρηση της θερμοκρασίας μέχρι την ολοκλήρωση των Openwhisk actions), στέλνεται η τιμή της μέσω του endpoint /metrics στο εργαλείο του Prometheus που μαζί με το εργαλείο για το visualization των metrics grafana, επιτρέπουν τη συνεχή παρακολούθηση της σε χρήσιμα γραφήματα..

Επιπρόσθετα, καθώς η γενικότερη παρακολούθηση της θερμοκρασίας ασχέτως αν ξεπερνά ή όχι το ορισμένο κατώφλι μπορεί να αποβεί χρήσιμη για ευνόητους λόγους, το Grafana παρέχει και λειτουργία MQTT Consumer, ώστε να παρακολουθείται η θερμοκρασία σε πραγματικό χρόνο.

Μοτίβα Σχεδιασμού και Μηχανισμοί

Pub/Sub Pattern

Όπως περιγράφηκε και στο κεφάλαιο της αρχιτεκτονικής, η εφαρμογή λειτουργεί κάτω από το μοτίβο σχεδιασμού Pub/Sub, καθώς εκμεταλλεύεται το πρωτόκολλο MQTT για την αποτελεσματική απορρόφηση των μετρήσεων θερμοκρασίας.

Συγκεκριμένα:

- επιτρέπεται η αδιάλειπτη επικοινωνία του αισθητήρα (publisher) και της εφαρμογής node-red (subscriber),
- τα δεδομένα μεταδίδονται σε πραγματικό χρόνο
- διευκολύνεται η επέκταση της εφαρμογής, καθώς μπορούμε, χωρίς να αλλάξουμε την λογική της αρχιτεκτονικής, να προσθέσουμε νέες συσκευές και νέους αισθητήρες που θα στέλνουν τα δεδομένα τους στα αντίστοιχα topics.

Retry Pattern

Όπως είδαμε παραπάνω, η λειτουργία των Openwhisk actions που αφορούν την έγκαιρη ειδοποίηση του χρήστη αλλά και την αποθήκευση των δεδομένων είναι ζωτικής σημασίας για την εφαρμογή, για αυτό και λειτουργούν κάτω από το πλαίσιο επανάληψης σε περίπτωση αποτυχίας τους τη πρωτη φορα.

Αυτό εξασφαλίζει στην εφαρμογή μεγαλύτερη αξιοπιστία και ανοχή σε σφάλματα, καθώς ελαχιστοποιεί το ρίσκο χαμένων ειδοποιήσεων και δεδομένων.

Μηχανισμοί Παρακολούθησης, Αυτοδιαχείρισης και Απόδοσης

Η εφαρμογή μέσω της χρησιμοποίησης του Grafana και του Prometheus επιτρέπει τη διαρκή παρακολούθηση των δεδομένων αλλά και της κατάστασης λειτουργίας της.

Η καταγραφή της καθυστέρησης κατά την εκτέλεση του service και η απεικόνιση της σε γραφήματα μας επιτρέπει να υπολογίζουμε τους χρόνους απόκρισης της εφαρμογής και να εντοπίζουμε τυχόν διακυμάνσεις στην ομαλή λειτουργία της.

Τέλος, οι μηχανισμοί επανάληψης που έχουν ήδη αναφερθεί αρκετά, είναι μια μορφή αυτοδιαχείρισης της εφαρμογής χωρίς την παρέμβαση του ανθρώπινου παράγοντα.

SLA

Το Συμβόλαιο Επιπέδου Υπηρεσίας βρίσκεται σε ξεχωριστό αρχείο.

Cumulative Distribution Function (CDF)

Χρησιμοποιούμε το Cumulative Distribution Function (CDF) για τον χρόνο ανταπόκρισης της υπηρεσίας μας. Αφού έχουμε συλλέξει 100 δείγματα του χρόνου καθυστέρησης της υπηρεσίας μας σε ένα αρχείο CSV, χρησιμοποιούμε το CDF για να ορίσουμε τον χρόνο ανταπόκρισης σύμφωνα με τον στόχο μας. Συγκεκριμένα, θέτουμε ως στόχο να διατηρούμε τον χρόνο ανταπόκρισης στα 6.000 χιλιοστά του δευτερολέπτου (6 δευτερόλεπτα) ή λιγότερο για όλες τις κρίσιμες λειτουργίες. Αυτό το ποσοστημόριο χρόνου ανταπόκρισης πρέπει να μην υπερβαίνεται για κάθε έναν μήνα του ημερολογίου, με την παρακολούθηση και αξιολόγηση να γίνεται μηνιαία.

latency.csv

column0	Rec
6,006	1
6,004	1
6,004	1
6,004	1
6,003	1
6,002	1
6,002	1
6,002	1
6,002	1
6,001	1
6,001	1
6,001	1
6,001	1
6,000	1
6,000	1
6,000	1
5,997	1
5,997	1
5,907	1
5,905	1
5,904	1
5,903	1

Πλεονεκτήματα και Μειονεκτήματα

Πλεονεκτήματα

Ένα από τα σημαντικότερα πλεονεκτήματα της εφαρμογής μας είναι η αποθήκευση δεδομένων στο MinIO, καθώς μας δίνει τη δυνατότητα να κάνουμε ιστορική ανάλυση της θερμοκρασίας και να μπορέσουμε με αυτόν τον τρόπο να εντοπίσουμε τάσεις, απειλές, να βελτιστοποιήσουμε τις στρατηγικές μας διατήρησης σταθερής θερμοκρασίας και να έχουμε αποδείξεις ότι η θερμοκρασία που διατηρούμε είναι σύμφωνη με τους κανονισμούς που ορίζονται.
Επιπλέον, με την συνεχή και αυτοματοποιημένη παρακολούθηση της θερμοκρασίας σε πραγματικό χρόνο, είμαστε ικανοί να βελτιστοποιήσουμε την κατανάλωση ενέργειας μειώνοντας με αυτόν τον τρόπο το κόστος και το περιβαλλοντικό αποτύπωμα.

Μειονεκτήματα

Ένα σημαντικό μειονέκτημα, που όμως αφορά μόνο την αρχική υλοποίηση, είναι η πολυπλοκότητα της εγκατάστασης όλου αυτού του συστήματος, η οποία εμπεριέχει τη ρύθμιση, διαμόρφωσή και επικοινωνία των επιμέρους components όπως είναι οι αισθητήρες, το Node-RED, το MinIO και το OpenWhisk.
Η ενσωμάτωση και η διατήρηση αυτής της αρχιτεκτονικής μπορεί να συνεπάγεται μεγάλο κόστος ως προς τους αισθητήρες και την ενεργειακή κατανάλωση, το οποίο μπορεί να αυξηθεί ακόμα περισσότερο αν χρειαστεί μελλοντικά να γίνει scale up. Ακόμα, μπορεί να είναι αναγκαία η πιστοποίηση για την χρήση εργαλείων και υπηρεσιών, η οποία οδηγεί σε επιπλέον κόστος.

Σύγκριση με σύγχρονες αρχιτεκτονικές

Παραδοσιακά HVAC Συστήματα

Σε σύγκριση με τα παραδοσιακά HVAC συστήματα, αυτή η αρχιτεκτονική προσφέρει πιο ακριβή έλεγχο της θερμοκρασίας, με αποτέλεσμα την εξοικονόμηση ενέργειας. Τα συστήματα HVAC είναι συχνά λιγότερο granular με αποτέλεσμα να μην ανταποκρίνονται τόσο γρήγορα στις διακυμάνσεις της θερμοκρασίας.

IoT-Based Λύσεις

Υπάρχουν παρόμοιες IoT-Based λύσεις, οι οποίες όμως ενδέχεται να μην περιλαμβάνουν τις severless υπολογιστικές δυνατότητες του OpenWhisk, σε αντίθεση με τη συγκεκριμένη αρχιτεκτονική που γίνεται triggered όποτε χρειάζεται με αποτέλεσμα να είναι πιο αποδοτική.